Proyecto final Sistemas Embebidos 2022

Graff Javier.[1] , Iurman Franco.[1] , Mazzarello Nicolas.[1] , Miguel Agustín[1].
[1]*Departamento de Ciencias e Ingeniería de la Computación, 8000 Bahía Blanca, Argentina.*

Resumen

Este proyecto consiste en controlar un vehículo a través de una pulsera inalámbrica, la cual contiene un giroscopio que obtiene los movimientos que se realicen con el brazo y luego enviar mediante radio frecuencia estos datos al vehículo. El vehículo cuenta también con un sensor de proximidad, el cual inhabilita el movimiento del mismo si hay algún objeto delante suyo y así evitar colisiones. Además, el vehículo cuenta con su propia red de wifi para poder conectarse al mismo y poder visualizar mediante una video cámara la visión del auto.

Descripción del problema, objetivos y motivación del trabajo

El desafío al que nos enfrentamos consistió en el desarrollo de un vehículo controlado mediante una pulsera de forma remota con las herramientas y los conocimientos provistos durante el cursado por la materia. Manejado a control remoto y con la posibilidad de ver en tiempo real donde se encuentra a través de una cámara web, nos permite abstraernos de la presencia física de un operario en zonas fuera del alcance de su visión.
La motivación fue, en gran parte, ver si podíamos utilizar dichos conocimientos dictados en la materia para implementar una solución real en un ambiente de desarrollo. Trabajando en equipo y asignando a cada uno de los integrantes una tarea específica, las cuales corresponden a partes individuales del vehículo se pudo obtener la versión final al unir cada una de ellas, poniendo así en marcha este vehículo.

Hardware y Software

Hardware:

- ➔ Vehículo:
  - Arduino UNO
  - Arduino NANO
  - Raspberry pi 4
  - Camara para raspberry pi 4 (Sony IMX219 de 8 megapíxeles)
  - Sensor de proximidad HC-SR04
  - Chasis (Motores, ruedas)
  - Baterias (power bank, 1 baterias 9v)
  - Receptor RF 433 MHz
  - Módulo controlador de Motor L298N
  - Motores Dc 3v a 6v
  -
- ➔ Pulsera:
  - Arduino Nano
  - Transmisor RF 433 MHz
  - Bateria 9v
  - Giroscopio/Acelerómetro MPU6050
  - Switch/pulsador

Software

- Arduino IDE
- Hostapd
- VNC: Conexión de escritorio remoto para Raspberry
- Conexión SSH con la Raspberry
- Hostapd
- Python 3

Solución adoptada

Protocolo de comunicación entre Arduino NANO y Arduino UNO

En la primera versión la información medida por el giroscopio fue tomada por el arduino NANO y enviada directamente de forma cruda al Arduino UNO en donde era procesada, sin embargo en la versión final se diseñó un protocolo para que el procesamiento se realice directamente en la placa Arduino NANO, y la comunicación conste de un solo byte que representa la orden, esta orden solo es enviada al ser pulsado un botón de la pulsera. Se optó por colocar este botón para no enviar valores no queridos a la Arduino Uno, con el fin de poder mover el brazo libremente y cuando realmente se quiera enviar una orden, haya que apretar el botón.

El protocolo consiste en las siguientes órdenes:

0	Detener el auto
1	Avanzar
2	Marcha atrás
3	Girar izquierda
4	Girar derecha

Se utilizó la librería RadioHead para enviar la orden del Arduino Nano al Arduino UNO, ya que esta misma nos permite cambiar parámetros de la liberia (elegir que utilice el timer2 antes que el timer1) que entraban en conflicto con otras librerías.

Link de la libreria: https://www.airspayce.com/mikem/arduino/RadioHead/

Protocolo de comunicación entre Raspberry pi y Arduino NANO
Este protocolo es mucho más sencillo ya que solo se debe comunicar si el auto tiene un obstáculo en frente o no.
El protocolo consiste en las siguientes órdenes:

'A'	Hay un obstáculo a menos de 30 cm
'E'	No hay un obstáculo a menos de 30 cm

Arduino UNO
En la placa Arduino UNO que se encuentra sobre la estructura del auto se realizan cuatro funciones fundamentales

- Lectura de la orden proveniente del arduino NANO que se encuentra en la pulsera,
- Lectura de la orden asociada al sensor de proximidad que proviene de la raspberry pi
- Procesamiento de toda la información recibida y
- El envío de órdenes hacia el módulo L298N el cual se encarga de controlar ambos motores DC instalados junto a las ruedas.

Para la recepción de señales de radiofrecuencia se utilizó una librería ya nombrada RadioHead. El programa se encarga de recibir constantemente el byte relacionado a la orden y almacenarlo.
Para la recepción de la orden correspondiente al sensor de proximidad constantemente se lee y almacena un carácter.
Para el procesamiento de la señal en primer lugar se analiza si hay un obstáculo frente al vehículo, de ser así se bloquea la opción de avanzar, caso contrario se pueden ejecutar todos los movimientos posibles. Para cada movimiento posible existe una función implementada que establecerá los valores adecuados en los pines correspondientes al controlador de motor.

Control del motor
El controlador L298N está compuesto de dos puentes H, uno para cada motor. Por un lado se establece si el motor está encendido o apagado (los pines ENA y ENB, uno para cada motor, los valores posibles van de 0 a 255 representando la velocidad) y por el otro se setean los pines IN1, IN2, IN3 e IN4, dos por cada motor, donde de acuerdo a cómo se configure, el motor marchará en un sentido o en el contrario, la configuración funciona de la siguiente manera:

	Adelante	Atrás	Freno
IN1 (o IN3)	HIGH	LOW	LOW
IN2 (o IN4)	LOW	HIGH	LOW

Arduino NANO
La placa Arduino NANO que se encuentra en la pulsera, es la encargada de obtener los valores que nos entrega el giroscopio. Los valores que utilizamos en el proyecto son los valores Ax y Ay que nos entrega el acelerómetro, con el valor Ax realizamos las funciones de "Detener el auto", "Avanzar" y "Marcha atrás", con el valor Ay realizamos las funciones "Girar izquierda" y "Girar derecha".

Los valores de Ax y Ay son un float, el cual representa los grados en los que se encuentra el giroscopio, según estos valores, codificamos las funciones mencionadas arriba:

Valores equivalentes a grados:

-4° < Ax < 4°	Detener el auto
Ax < -4°	Avanzar
Ax > 4°	Marcha atrás
Ay < -4°	Girar izquierda
Ay > 4°	Girar derecha

Luego de procesar estos valores, si se presiona el botón enviaremos mediante el módulo de radio frecuencia la orden al Arduino UNO.

Raspberry pi 4

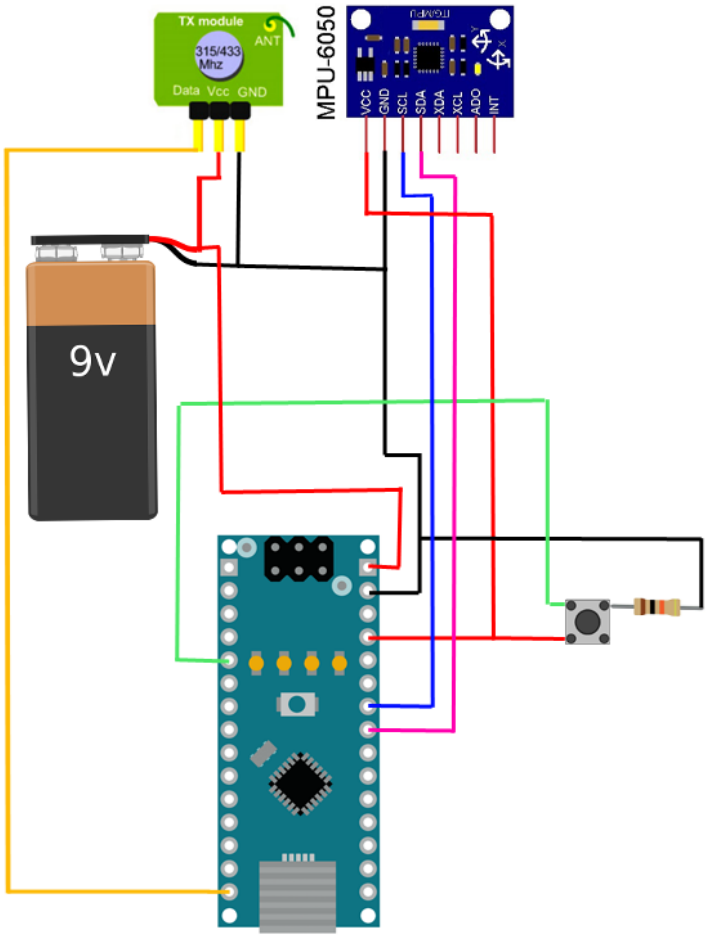
En esta plataforma desarrollamos el streaming de video tomando por la webcam y la obtención, procesado y transmisión de los datos obtenidos por el sensor de proximidad. Para la parte de video, en primer lugar se generó un punto de acceso para una red WiFi llamada “WIFI AUTO” y dentro de esa red se corrió un servidor que se encargó de tomar los datos de video y mostrarlos. La decisión de implementar una propia red en lugar de realizar un streaming por medio de alguna plataforma o un servidor corriendo en internet se tomó con el objetivo de no tener que depender de estar en cercanía de un router con conexión a internet para poder ver el video si no que con solo estar cerca del rango de alcance de la red WiFi, cualquier persona con un dispositivo pueda conectarse directamente con el auto. El desarrollo del servidor que muestra el video fue desarrollado en *Python* y la herramienta usada para que la tarjeta de red actúe como punto de acceso fue *Hostapd*.

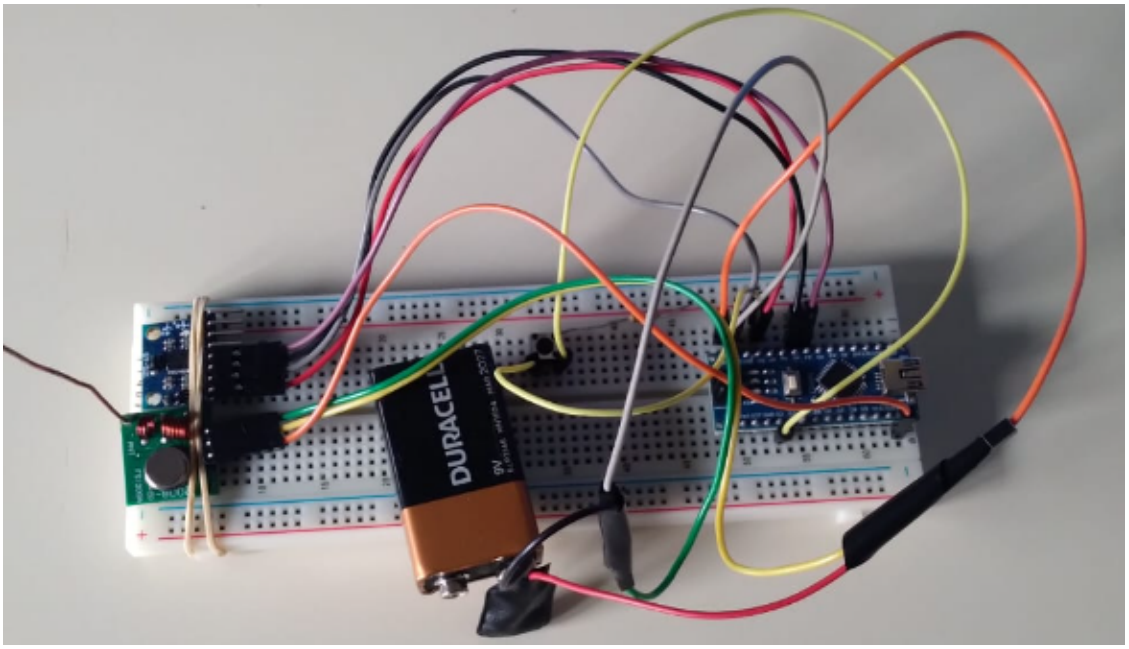
Para la parte del sensor de proximidad en primer lugar debimos solucionar el problema de que el sensor de proximidad opera a 5V mientras que las entradas y salidas del raspberry operan a 3.3V por lo que se realizó esta reducción de voltaje con un divisor de tensión. Con un programa desarrollado en *Python* se leen los datos provenientes del sensor, luego se procesan para obtener la medida en metros, finalmente se sigue un protocolo básico que envía un mensaje con una letra ‘E’ si la distancia medida es mayor a 30cm o la letra ‘A’ si la distancia medida es menor o igual a ese mismo umbral. La comunicación de los datos se realiza mediante el puerto serial entre el USB tipo A de la raspberry y el USB tipo B de la placa arduino UNO. Un programa corriendo en esta última es quien se encarga de tomar los datos por el serial y decidir si el auto debe avanzar o no.

Sobre esta placa corre un sistema operativo por lo tanto se pudo aprovechar la posibilidad de correr más de un programa en simultáneo y la posibilidad de realizar el debugging de una forma más sencilla.

En un principio se intentó utilizar el sensor de proximidad conectado a la placa Arduino UNO leyendo directamente la información desde sus pines de entrada/salida, sin embargo la forma de realizar las mediciones implicaba el uso de uno de los tres timers del arduino, los cuales ya estaban siendo utilizados para otras funciones. Es por eso que la solución adoptada consistió en sensar los datos desde la placa Raspberry Pi y comunicar esas mediciones a través de comunicación Serial. En la práctica los retardos de comunicación fueron imperceptibles por lo que la solución funcionó de acuerdo a lo planificado.

Anexo 1 Interconexión pulsera





Anexo 2 interconexión chasis del auto